

## Contents

<b>1</b>	<b>Routine/Function Prologues</b>	<b>2</b>
1.0.1	getcmap.F90 (Source File: getcmap.F90) . . . . .	2
1.0.2	cmapfile (Source File: getcmap.F90) . . . . .	4

# 1 Routine/Function Prologues

## 1.0.1 getcmap.F90 (Source File: getcmap.F90)

Opens and reads global precipitation forcing

CTIME = Current time

FTIMENRL = Nearest future data for NRL data

FTIMEHUFF = Nearest future data for HUFFMAN data

FTIMEPERS = Nearest future data for PERSIANN data

### REVISION HISTORY:

17 Jul 2001: Jon Gottschalck; Initial code

10 Oct 2001: Jon Gottschalck; Modified to adjust convective precip  
using a ratio of the model convective / total ratio

30 Jul 2002: Jon Gottschalck; Added PERSIANN and HUFFMAN global observed precip data sources

### INTERFACE:

```
subroutine getcmap()
```

### USES:

```
use lisdrv_module, only : lis, gindex
use time_manager
use time_module, only : tick
use cmapdomain_module, only : cmapdrv, def_cmap_ip_input
implicit none
```

### CONTENTS:

```
    endtime_cmap = 0
!-----
! Set parameter to measure 1.5 hour time offset when using HUFFMAN
!-----
    gap = 0.0001712328767098370
!-----
! Determine required observed precip data times
! (current, accumulation end time)
! Model current time
!-----
    yr1 = lis%t%yr  !current time
    mo1 = lis%t%mo
    da1 = lis%t%da
    hr1 = lis%t%hr
    mn1 = lis%t%mn
    ss1 = 0
    ts1 = 0
    call tick( ctime, doy1, gmt1, yr1, mo1, da1, hr1, mn1, ss1, ts1 )
!-----
```

```

! CMAP product end time
!-----
yr5 = lis%t%yr !end accumulation time data
mo5 = lis%t%mo
da5 = lis%t%da
hr5 = 6*(lis%t%hr/6)
mn5 = 0
ss5 = 0
ts5 = 6*60*60
call tick( ftime_cmap, doy5, gmt5, yr5, mo5, da5, hr5, mn5, ss5, ts5 )
if(ctime > cmapdrv%griduptime1 ) then
  print*, 'changing cmap grid to 2002 - '
  cmapdrv%ncold = 768
  cmapdrv%nrold = 384
!-----
! Reinitialize the weights and neighbors
!-----
  kgdsi = 0
  kgdsi(1) = 4
  kgdsi(2) = 768
  kgdsi(3) = 384
  kgdsi(4) = 89462
  kgdsi(5) = 0
  kgdsi(6) = 128
  kgdsi(7) = -89462
  kgdsi(8) = -469
  kgdsi(9) = 469
  kgdsi(10) = 192
  kgdsi(20) = 255
  call def_cmap_ip_input(kgdsi)
  cmapdrv%gridchange1 = .false.
endif
!-----
! Ensure that data is found during first time step
!-----
  if ( lis%f%gpcpsrc.eq.4.and. get_nstep().eq. 1 ) endtime_cmap = 1
!-----
! Check for and get CMAP CPC Precipitation data
!-----
  if (lis%f%gpcpsrc==4) then
    if ( ctime > cmapdrv%cmaptime ) endtime_cmap = 1
    if ( endtime_cmap == 1 ) then !get new time2 data
      ferror_cmap = 0
      call cmapfile( name, cmapdrv%cmapdir, yr5, mo5, da5, hr5 )
      print*, 'Getting new CMAP CPC precip data',name
      call glbprecip_cmap( name, ferror_cmap, hr5 )
      cmapdrv%cmaptime = ftime_cmap
    endif !need new time2
  endif

```

```

endif
return

```

---

## 1.0.2 cmapfile (Source File: getcmap.F90)

This subroutine puts together CMAP file name for 6 hour file intervals

INTERFACE:

```

subroutine cmapfile( name, cmapdir, yr, mo, da, hr)

```

CONTENTS:

```

91 format (a4)
92 format (80a1)
93 format (a80)
94 format (i4, i2, i2, i2)
95 format (10a1)
96 format (a40)
97 format (a10)
98 format (a1, i4, i2, a1)
99 format (8a1)
!-----
! Make variables for the time used to create the file
! We don't want these variables being passed out
!-----
  uyr = yr
  umo = mo
  uda = da
  uhr = 6*(hr/6) !hour needs to be a multiple of 6 hours
  umn = 0
  uss = 0
  ts1 = -24*60*60 !one day interval to roll back date.

  open(unit=90, file='temp', form='formatted', access='direct', recl=80)
  write(90, 96, rec=1) cmapdir
  read(90, 92, rec=1) (fbase(i), i=1,80)

  write(90, 98, rec=1) '/', uyr, umo, '/'
  read(90, 99, rec=1) fdir
  do i = 1, 8
    if ( fdir(i) == ' ' ) fdir(i) = '0'
  end do

  write(90, 97, rec=1) 'cmap_gdas_'
  read (90, 92, rec=1) (fsubs(i), i=1,10)

```

```
write(90, 94, rec=1) uyr, umo, uda, uhr
read(90, 95, rec=1) ftime
do i = 1, 10
  if ( ftime(i) == ' ' ) ftime(i) = '0'
end do

write(90, 94, rec=1) uyr, umo, uda, uhr
read(90, 95, rec=1) ftime
do i = 1, 10
  if ( ftime(i) == ' ' ) ftime(i) = '0'
end do

write(90, 91, rec=1) '.grb'
read (90, 92, rec=1) (fsubs2(i), i=1,4)
c = 0
do i = 1, 80
  if ( (fbase(i) == ' ') .and. (c == 0) ) c = i-1
end do

write(90, 92, rec=1) (fbase(i), i=1,c), (fdir(i), i=1,8), &
                    (fsubs(i), i=1,10),(ftime(i), i=1,10), &
                    (fsubs2(i), i=1,4)

read(90, 93, rec=1) name

close(90)
return
```